COSC2425-002 Group Project 2:

Controlling LED Output with an IR Range Detector

Olaf Alexander, Benjamin Holt, Stacy Bridges

Austin Community College

Fall 2014

Table of Contents

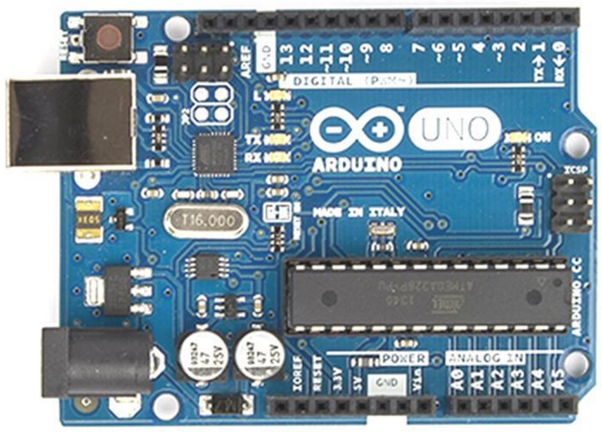
Introduction

For Group Project 2, our team combined an Arduino Uno board with a distance measuring sensor and a bright-white LED (light emitting diode). Our goal was to conduct an experiment to see if we could load assembly code onto the Arduino and then use it to control the brightness of the LED based on voltage fluctuations emitted by the distance sensor.

To demonstrate our experiment, we mounted the LED on a short demonstration breadboard and then observed the LED to dim or brighten while varying the distance sensor's proximity to various objects.

Additional details around the components and code we used for this project have been detailed in the sections that follow.

The Microcontroller Platform



Our project uses the Arduino Uno to interface a distance sensor to a bread board loaded with an LED. The Arduino is the perfect tool for a project of this size, providing a microcontroller board based on the ATmega328. The board provides 14 digital input/output pins, 6 analog inputs, and a USB connection for easy programming and power supply from a computer desktop. One nice feature provided by the Arduino board is that 6 of its digital input/output pins can be programmed to pull double-duty as Pulse Width Modulation (PMW) pins. For our project, this feature provided a nice solution for interfacing our data-sensor input to our LED.

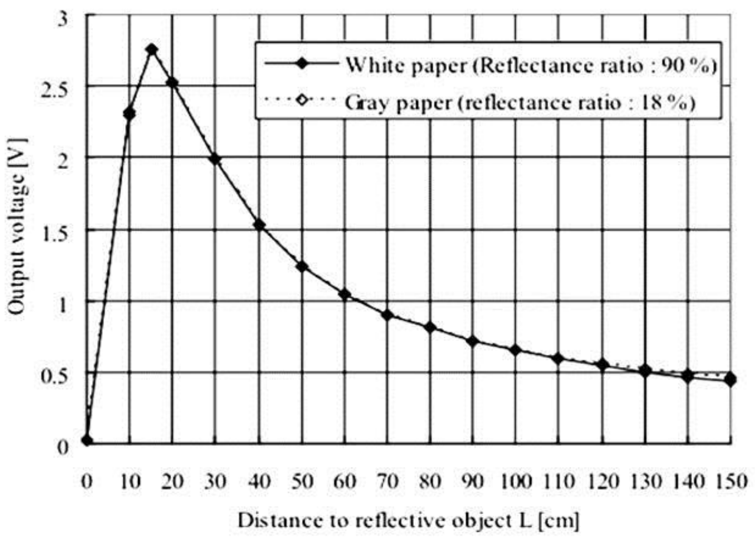
The data-sensor that we used for this project is also perfectly matched to the Arduino microcontroller, which allowed our team to focus more on the assembly language and less on problem solving where component compatibility is concerned. For example, the sensor we used operates on 4.5 to 5.5 V, which is nicely aligned to the Arduino’s resident 5 V output, and the sensor also outputs power at 33 mA, which is nicely aligned to the 40 mA capability provided at each input/output pin of the Arduino board.

The Test Device

The sensor we are using for our project is the GP2Y0A02YK0F by Sharp. This is a distance measuring unit capable of measuring between 20 cm and 150 cm via an IRED (infrared emitting diode) and a triangulation method.

The sensor operates on 4.5 to 5.5 V and 33 mA. It uses internal circuitry to produce an analog voltage variance as an output. As the distance of an object from the sensor increases, the voltage output by the sensor decreases. (https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf).

Because the voltage varies in relation to the distance of an object from the sensor, it would be possible to feed the sensor to a comparator and create a threshold detection mechanism. However, we decided to take this a step further. Rather than use a comparator, the output of the sensor will be fed to an analog-to-digital converter within the Arduino Uno. This will then produce a number that varies in relation to the distance an object is from the sensor. This number can then be used by the micro-controller to vary the intensity of an LED.



Because this sensor operates with an IRED, there are a couple of important operation and usage notes that we had to be aware of. The lens must be kept clean, however it should not be washed. Also, the reflectively of the object being measured could have an effect on the results. If the measured object is in motion, the moving direction for the object and the line between emitter center and detector center should be vertical. This applies to a direction of motion that is perpendicular to the unit normal of the sensor plane. An object that remains within the measurable field of the sensor can reliably move closer to and farther from the sensor.

Development tools

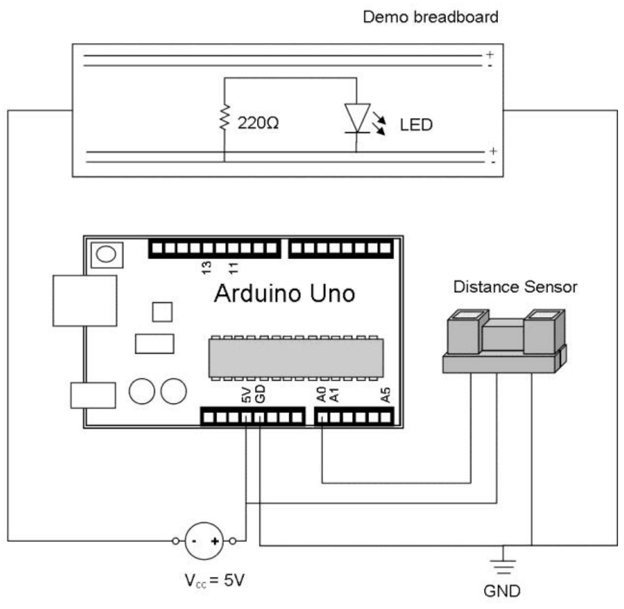
The additional components that were required to complete this project were minimal, as listed below.

- Bright-white LED
- 220 Ω Ohm resistor
- Mini bread board
- Jumpers

Your Experiment

To set up the experiment, we attached the distance sensor to one of the Arduino's analog inputs, installed an LED and 220 Ω resistor onto a small demonstration breadboard, and then supplied voltage to the LED via the Arduino's resident 5V output pin.

For the control program, we began with a simple C++ script in the Arduino's IDE (integrated development environment), and then we worked with the voltage readings in the Serial Monitor until achieving a reasonable interaction between the LED and the distance sensor. Based on that proof of concept, we began re-tooling the code to incorporate as much assembly language as possible given our current proficiencies in the language. For the final code set, we settled on a hybrid approach that uses assembly language for the pin mapping and analog input, but uses C++ for writing PWM output to the LED pin.



Conclusions

While the primary purpose of this lab experiment was to learn more about how assembly code may be used to interface with a sensor, we actually also learned more about the compilation structure of the code files that Arduino constructs “under the hood.” By toggling the verbosity of the IDE (integrated development environment), we were able to locate and analyze the source files directly. We can see this skill set proving helpful in future projects where efficiency may come in the form of reducing reliance on the IDE in favor of coding and compiling directly from the command line.

Contributions

All team members wore several hats over the course of this project. Some of the specific contributions are called out below, including those made at the group level as well as the individual level.

Special recognition goes to: Benjamin, for getting the initial research completed for the distance sensor; Olaf, who did the lion's share of initial research on the code that was needed to transform the original C++ program into a workable assembly program; and Stacy, who managed the content rodeo necessary to stand up the initial graphics and layout for the final report.

- **Olaf Alexander**
  - Hardware custodian
  - Breadboard engineer
  - Code specialist
- **Benjamin Holt**
  - Meeting coordinator
  - Distance sensor research
- **Stacy Bridges**
  - Document coordinator
  - Arduino platform research
- **All**
  - Report authoring
  - Testing

Project code

The code below is the assembly-language code that our team wrote in order to interface the data-sensor device to the LED output. This assembly code sets up the Arduino device by mapping the input pins and reading the data-sensor's analog inputs.

```
as_read:
        ldi     r24,   3
        sts     0x7C,  r24
        ldi     r24,   0x87
        sts     0x7A,  r24

Wait:
        lds     r24,   0x7A
        sbrs    r24,   4
        rjmp    Wait

        ldi     r24,   0x78
        ldi     r25,   0x79
        lsr     r24
        lsr     r24
        lsl     r25
        lsl     r25
        add     r24,   r25
        ldi     r25,   0xFF
        sub     r25,   r24

        ret
```